Assignment 4 - CSC/DSC 265/465 - Spring 2019 - Due April 23

Q1: This question will make use of data downloaded from GEO (Gene Expression Omnibus) identified by series accession number GSE364. From an abstract describing the study:

> We analyzed the expression profiles of HCC [hepatocellular carcinoma] samples without or with intra-hepatic metastases. Using a supervised machine-learning algorithm, we generated for the first time a molecular signature that can classify metastatic HCC patients and identified genes that were relevant to metastasis and patient survival [Ye et al (2003) "Predicting hepatitis B virus-positive metastatic hepatocellular carcinomas using gene expression profiling and supervised machine learning." *Nature Medicine* (4):416–423].
>
> Original data can be found at `https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE364`

The file `GSE364n50.csv` contains 50 gene expression profiles from the GSE364 data set. Use the following command

```
gem = read.table(file='GSE364n50.csv',sep=',',row.names=1,header=F)
```

Then `gem` will be a data frame with 50 rows, each representing the gene expression profile of a single sample. There are 5913 columns, each representing the expression of a particular gene. The gene names themselves are not included here, and can simply be labelled 1 - 5913. The `R` expression `row.names(gem)` gives labels for the samples themselves. Otherwise, every element in `gem` is a numerical gene expression level.

Finally, two classes are represented among the 50 gene expression profiles. *Metastasis* occurs when, for example, a cancerous tumor spreads to a new site. Samples represented in rows 1-30 are from metastatic tissue, while rows 31-50 are from non-metastatic tissue. The problem considered here is to build a classifier with which tissue can be classified as metastatic or non-metastatic based on gene expression profiles (which therefore constitute the feature set). Therefore, you will have to create a factor variable which classifies each row accordingly.

(a) Use the `prcomp` function (from library `stats`) to create a matrix of principal components, using the gene expressions as a feature set. Use centering, but not scaling. Then construct a QDA classifier for metastasis class, using the first four principal components as features. Use the `CV=TRUE` option. Also, specify a uniform prior distribution for the classes (to do this, use option `prior=c(0.5,0.5)`).

  (i) If do not specify the prior probabilities, what values will be used?

  (ii) The object output from the function `qda` contains a matrix giving the posterior probability of each class (the predicted class is, of course, the one with the highest posterior probability). Create a single vector giving the maximum posterior probability p_{max} for each observation. What is the correct classification rate for observations with $p_{max} \geq 0.75$, and for observations with $p_{max} < 0.75$? Use a Wilcoxon rank-sum test to determine whether or not there is a difference in the distribution of p_{max} between observations that were correctly classified and those that weren't. Report a P-value.

  (iii) Create a grid of pairwise scatterplots for the first four principal components, using the `pairs` function. Use colors black and red for metastasis -ve and +ve observations, respectively. In addition, use plotting character "+" (`pch=3`) for correctly classified observations, and "○" (`pch=1`) for incorrectly classified observations. In general terms, how do the correctly and incorrectly classified observations differ graphically?

(b) We will next use cross-validation to determine the number of principal components to include in the classifier. We take the model parameter to be K if the first K principal components are used. We use two methods:

- The PCA is done first, using the entire data set. The principal components are accepted as the new feature set. Then K is varied using values $1, 2, \ldots, 17, 18$. For each K, LOOCV is used to evaluate a QDA classifier accepting the first K principal components as the feature set.

- Again, K is varied using values $1, 2, \ldots, 17, 18$. For each K, LOOCV is applied. The PCA is recalculated for each new training data set considered. Whenever principal components are calculated for test data, the loadings used are those calculated using the training data only.

Obtain classification error rates for each K using both CV methods. Use the `matplot` function to plot classification error against K, placing both methods on the same graph. How do the methods compare? What would be the recommended number of principal components K?

Q2: In order to assess the accuracy of a classification algorithm, we can simply compare known classifications to the predictions. On the other hand, clustering is an unsupervised learning problem. The output is not a class prediction but a partition (into clusters). We can still assess the accuracy of a clustering algorithm if it is tested on data for which some true clustering structure exists, and is known. In this case, the suitable accuracy metric would be a *partition distance* or *partition metric*, applied to the partitions induced by the known clustering, and the clustering output by the algorithm.

For example, consider two partitions of $\{1, 2, 3, 4\}$, $Q = (1, 2 \mid 3, 4)$ and $R = (1, 2 \mid 3 \mid 4)$. Here Q contains 2 clusters, while R contains 3 clusters. We can define partition distance $D(Q, R)$ as the minimum number of elements which need to be removed in order for the remaining partitions to be equal. If we remove element 4 from Q and R, then this leaves partitions $Q' = (1, 2 \mid 3)$ and $R' = (1, 2 \mid 3)$, which are equal, so that $D(Q, R) = 1$. On the other hand, if $S = (1 \mid 2 \mid 3 \mid 4)$, then $D(Q, S) = 2$.

We will consider here mutual information as a partition distance. Denote the collection of m labels $\mathcal{I}_m = \{1, \ldots, m\}$. Let $P = (p_1, \ldots, p_m)$ be a probability distribution on $\mathcal{I}_m$. The entropy of P is defined as

$$H(P) = -\sum_{i=1}^{m} p_i \log_b(p_i), \tag{1}$$

for any fixed logarithm base $b > 1$. Unless otherwise specified we omit the base b, and we may take $\log = \log_b$ to be the natural logarithm. Furthermore, we adopt the convention $0 \cdot \log_b(0) = 0$, noting that $\lim_{p \downarrow 0} p \log_b(p) = 0$, thus ensuring continuity at $p = 0$.

Next, suppose we are given a joint probability distribution P_{AB} defined on two finite sets $A = \{1, \ldots, n\}$, $B = \{1, \ldots, m\}$. This can be represented in tabular form:

Table 1: Random selection from A and B

		B				
		1	2	$\ldots$	m	
A	1	p_{11}	p_{12}	$\cdots$	p_{1m}	α_1
	2	p_{21}	p_{22}	$\cdots$	p_{2m}	α_2
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	n	p_{n1}	p_{n2}	$\cdots$	p_{nm}	α_n
		β_1	β_2	$\ldots$	β_m	1

Here

$$\begin{aligned}\alpha_i &= \sum_{j=1}^{m} p_{ij}, \;\; i = 1, \ldots, n \\ \beta_j &= \sum_{i=1}^{n} p_{ij}, \;\; j = 1, \ldots, m\end{aligned}$$

define the marginal distributions $P_A = (\alpha_1, \ldots, \alpha_n)$, $P_B = (\beta_1, \ldots, \beta_m)$. The entropy for P_{AB} uses the above definition, but requires double summation:

$$H(P_{AB}) = -\sum_{i=1}^{n} \sum_{j=1}^{m} p_{ij} \log_b(p_{ij}).$$

Mutual information is then defined as

$$MI(A,B) = H(P_A) + H(P_B) - H(P_{AB}).$$

It can be shown that the following bounds always hold

$$0 \leq MI(A,B) \leq \min\{H(P_A), H(P_B)\}.$$

Suppose Q and R are two partitions of N samples, labeled $\{1, \ldots, N\}$. Label the clusters in Q and R $i \in A = \{1, \ldots, n\}$ and $j \in B = \{1, \ldots, m\}$. Then the two partitions are equal if any cluster in one partition is exactly equal to a cluster in the other partition (in this case we would have $n = m$). The labeling makes no difference.

Let n_{ij} be the number of samples that are in both the ith cluster of Q and the jth cluster of R. Then let

$$p_{ij} = \frac{n_{ij}}{N}.$$

If we then calculate $H(P_A)$, $H(P_B)$ and $MI(A,B)$ for this distribution, we can define a clustering similarity index

$$I(Q,R) = \frac{MI(A,B)}{\max\{H(P_A), H(P_B)\}},$$

with larger values indicating similar clusterings.

(a) Install and load the package `partitionComparison`. This is an S4 class package, and contains the function `mutualInformation`, which calculates a partition distance based on mutual information in the manner described above. It accepts two `partition` class objects, which must first be created, using vectors of partition labels. For example, to calculate the mutual information partition distance between $Q = (1,2 \mid 3,4)$ and $R = (1,2 \mid 3 \mid 4)$ use code:

```
> pp = new("Partition", c(1,1,2,2))
> qq = new("Partition", c(1,1,2,3))
> mutualInformation(pp,qq)
[1] 0.6931472
>
```

To calculate $H(P_A)$ or $H(P_B)$, the entropy of the distribution of a single partition, say `pp`, use command `mutualInformation(pp,pp)` (see question **Q4**).

(b) Create a hierarchical cluster of the `iris` data using the `hclust` function, using the 4 morphological measurements as the feature set, and all 150 observations. Use option `method="average"`.

(c) Use the cutree function to obtain from the resulting dendogram partitions with K clusters, for $K = 2, 3, \ldots, 9, 10$. For each K calculate the cluster distance between each pair of clusters (this should be based on the average agglomeration method). Then for each K determine the smallest of these cluster distances.

(d) For each K calculate the clustering similarity index $I(Q, R)$ for the true species clustering and the size K dendogram clustering.

(e) Plot the minimum cluster distance of Part (c) as a function of K. Then plot the clustering similarity index $I(Q, R)$ as a function of K. What do these plots suggest about the ability of hierarchical clustering to detect clustering structure, and about the accuracy of the clustering method?

Q3: For this problem use data set UScereal from the MASS package. The rows represent brands of breakfast cereal. The command rownames(UScereal) gives the brand names. Columns 2-8 and 10 give various quantitative measures of the ingredients in a single serving. Use these 8 columns to define a new feature matrix. Apply the transformation $f(x) = log_{10}(x + 1)$ to all values in the matrix.

(a) For $K = 1, 2, \ldots, 14, 15$ use function kmeans to create a clustering solution of size K. Use option nstart = 100. Create a vector which stores the within-cluster sum of squares SS_{within} for each value of K. Then create two plots, one which plots SS_{within} against K, and one which plots $R^2 = 1 - SS_{within}/SS_{total}$ against K. What do the plots suggest about the number of actual clusters?

(b) Suppose we accept as the true number of clusters the number K at which R^2 is not improved by more than 0.1 when one additional cluster is permitted. Then create a separate list of brand names for each cluster. Do these lists suggest that the clustering might be informative? Any sensible answer will be considered correct.

Q4: [For Graduate Students] This question refers to the partition distance discussed in question **Q2**.

(a) Verify that the mutual information can be written:

$$MI(A, B) = -\sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij} \log_b \left(\frac{\alpha_i \beta_j}{p_{ij}} \right).$$

(b) Verify the lower bound

$$MI(A, B) \geq 0.$$

HINT: Let (I, J) be a pair of random chosen classes, with $P((I, J) = (i, j)) = p_{ij}$. Write the expression of Part (a) as an expectation of the form

$$MI(A, B) = E\left[-\log_b(g(I, J))\right].$$

Then use Jensen's Inequality, which states that if $f(x)$ is a convex function, then

$$E[f(X)] \geq f(E[X])$$

for any random variable X for which the expectation is defined.

(c) Verify the upper bound

$$MI(A,B) \leq \min\{H(P_A), H(P_B)\}.$$

HINT: Write an expression for $H(P_A) - H(P_{AB})$ similar to the one given in Part (a). Verify that this expression is nonpositive. Do the same for $H(P_B) - H(P_{AB})$.

(d) Verify that the lower bound of $MI(A,B)$ is attained when the random selections from A and B are independent, that is, $p_{ij} = \alpha_i \beta_j$ for all i, j.

(e) Verify that the upper bound of $MI(A,B)$ is attained under either of the following two conditions:

i) Each row of Table 1 has exactly one nonzero entry.

ii) Each column of Table 1 has exactly one nonzero entry.

Furthermore, under condition i) we have $H(P_B) \leq H(P_A)$, and under condition ii) we have $H(P_A) \leq H(P_B)$, with strict inequality if $n \neq m$, and equality if $n = m$.

(f) Note that the clustering similarity index

$$I(Q,R) = \frac{MI(A,B)}{\max\{H(P_A), H(P_B)\}}$$

uses $\max\{H(P_A), H(P_B)\}$ in the denominator to normalize comparisons, rather than $\min\{H(P_A), H(P_B)\}$. It might seem more intuitive to use $\min\{H(P_A), H(P_B)\}$, since this is a strict upper bound of $MI(A,B)$.

Suppose Q is any partition of N labels with strictly less than N clusters. Let R be the partition which assigns exactly one observation into each of N clusters. Show that one of the conditions of Part (e) is satisfied. How does this example suggest that $\max\{H(P_A), H(P_B)\}$ is the better choice for normalizing comparisons?